

УДК 004.42

РЕАЛИЗАЦИЯ КЛИЕНТА SKYPE С СОБСТВЕННЫМ ВСТРОЕННЫМ ЯЗЫКОМ КОМАНД И ПОДДЕРЖКОЙ НАСТРАИВАЕМЫХ КЛАВИАТУРНЫХ КОМБИНАЦИЙ

Обедин Н.В.,**научный руководитель канд. техн. наук, доцент Кузьмин Д. А.*****Сибирский федеральный университет******Институт космических и информационных технологий***

Skype — бесплатное проприетарное программное обеспечение с закрытым кодом, обеспечивающее зашифрованную голосовую связь и видеосвязь через Интернет между компьютерами (VoIP), используя технологии пиринговых сетей, а также платные услуги для звонков на мобильные и стационарные телефоны. Несмотря на то, что на сегодняшний день Skype является самым популярным средством для общения в сети Интернет и совершения международных звонков [1], у его официального клиента существуют некоторые недостатки:

1. Закрытость исходного кода: согласно исследованиям Carnegie Mellon University [2], в закрытом программном обеспечении количество ошибок на 1000 строк кода в 100 раз больше, чем в открытом (20 против 0.17). Также в открытых программах быстрее реализуются новые функции [3].
2. Ограниченность в настройке и персонализации практически не позволяет модифицировать внешний вид и поведение клиента, а недостаток встроенных команд делает невозможным расширение функционала без изменения и перекомпиляции программы, что в совокупности с первым пунктом означает отсутствие возможности изменения работы клиента как таковой.
3. Слабое использование возможностей управления программой с помощью клавиатурных комбинаций: для часто используемых функций, например, поиска контакта или изменения статуса клавиатурные комбинации не назначены вообще; для тех же действий, которым назначены комбинации, изменить их невозможно.
4. Значительные различия во внешнем виде и функциональности клиента в зависимости от операционной системы при заявленной кроссплатформенности

Все вышеизложенные проблемы были решены созданием альтернативного открытого клиента с поддержкой собственного языка команд на базе SkypeKit API и фреймворка Qt.

SkypeKit состоит из двух частей: исполняемого файла, скомпилированного под различные платформы и исходного кода динамической библиотеки. Исполняемый файл представляет из себя клиент для сети Skype, который реализует специальный протокол, предоставляющий доступ к функциям сети для сторонних клиентов. Динамическая библиотека является API к этому протоколу. Принцип взаимодействия компонентов SkypeKit схематично показан на рис. 1.



Рисунок 1 – Принцип работы SkypeKit

Доступ к функциям API происходит с помощью наследования от специальных классов, предоставляющих абстракцию над элементами сети (например, Account или Conversation) и вызова их методов. Внешние события, такие как, например, входящий звонок или сообщение обрабатываются с помощью функции обратного вызова (Callback). Для эффективного использования памяти в SkypeKit встроен автоматический сборщик мусора (Garbage Collector).

Qt — кросс-платформенный инструментарий разработки ПО на языке программирования C++. Позволяет запускать написанное с его помощью ПО в большинстве современных операционных систем путём простой компиляции программы для каждой ОС без изменения исходного кода. Qt является полностью объектно-ориентированным, легко расширяемым и поддерживающим технику компонентного программирования. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Отдельно стоит отметить QtDeclarative — модуль, предоставляющий декларативный фреймворк для создания динамичных, настраиваемых пользовательских интерфейсов на языке QML, который, в свою очередь, основан на JavaScript. Это позволяет эффективно разделить разработку программы: основные функции и алгоритмы используют C++, в то время как код, отвечающий за интерфейсную часть написан на QML/JavaScript.

Клиент представляет из себя систему, состоящую из основного исполняемого файла, динамической библиотеки и исполняемого файла SkypeKit, а также ключа доступа pair.rem, необходимого для авторизации SkypeKit на серверах Skype и получения доступа к сети Skype. Для запуска системы требуется:

1. Запустить исполняемый файл SkypeKit
2. Подключиться по протоколу TCP/IP к локальному серверу, созданному SkypeKit
3. Зарегистрироваться на сервере Skype с помощью ключа доступа

Вышеописанные действия производятся автоматически, соответствующие функции запрограммированы в основном исполняемом файле. Если выполнение этих действий оказывается успешным, то создается объект класса Spyke, который реализует базовые функции по работе с API. Надо отметить, что данный объект является синглтоном, то есть создается один раз за время работы программы. Его уничтожение автоматически дает команду для завершения исполняемого файла SkypeKit. После создания этого объекта также создаются виджет для отображения интерфейса пользователя и специальный объект SpykeDispatcher, который связывает сигналы/слоты интерфейса с синглтоном Spyke. В завершение запускается основной цикл приложения и на экране появляется окно авторизации пользователя (рис. 2).

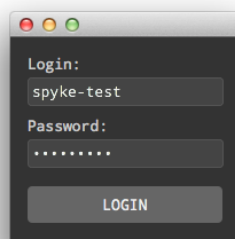


Рисунок 2 – Окно авторизации пользователя

После успешной авторизации перед пользователем предстает основное окно программы, содержащее списки контактов/конференций и сообщений (рис. 3).

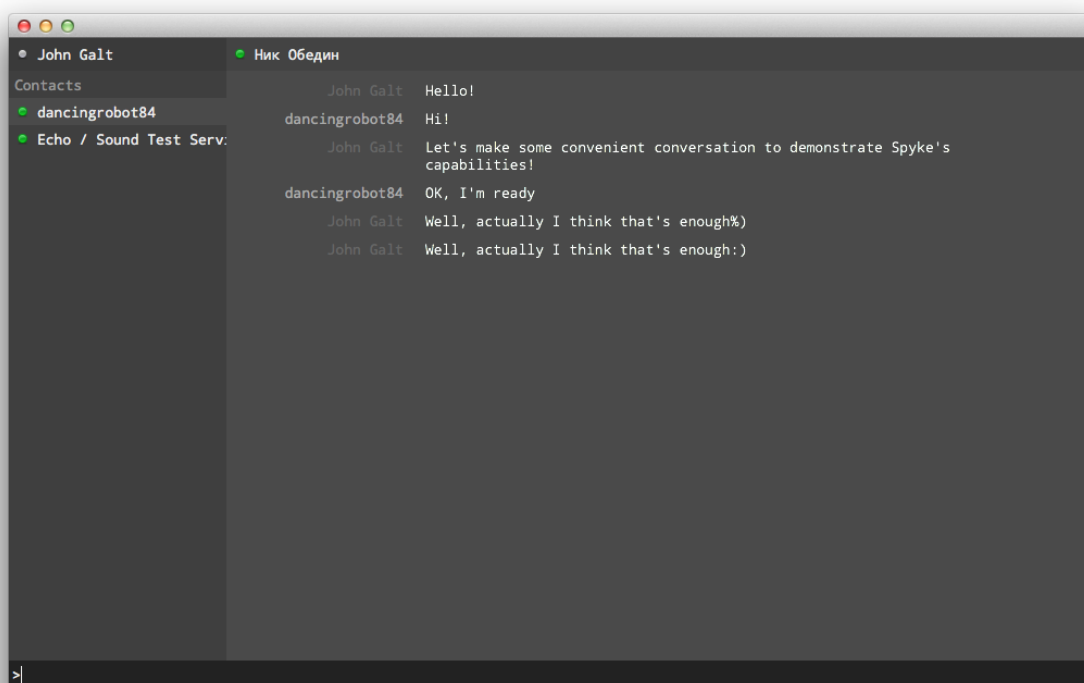


Рисунок 3 – Основное окно программы

Контакт-лист представляет собой список контактов и последних конференции пользователя. Как и в оригинальном клиенте реализована возможность видеть статус присутствия конкретного контакте, а также оповещение о новом сообщении в диалоге или конференции. Основу контакт-листа представляет класс `QContactListModel`, который является наследником `QAbstractItemListModel`. Этот класс реализует принятую в среде фреймворка Qt структуру разделения данных и их представления Model-View. Внутри этого класса для слежения за изменением статуса используются список контактов и список конференции, каждый элемент которых связан с моделью соответствующими сигналами. При регистрации изменения модель, в свою очередь, посылает сигналы виджету, который ее использует (в данном случае им является QML элемент `ListView`). Это происходит автоматически, что позволяет

предотвратить написание разработчиком большего количества кода.

При выборе контакта/конференции из контакт-листа в список сообщений загружаются сообщения из выбранного(и) контакта/конференции. Модель сообщений представляет собой класс `QMessageListModel`, который также наследуется от `QAbstractItemListModel`. Также, как и в случае с контакт-листом, надлежащей структурой данных здесь является список, но уже список сообщений.

Командная строка вместе с движком для обработки команд, перехватчиком клавиатурных комбинаций и виджетами списков контактов и сообщений написана на `QML/JavaScript` и выполняется во встроенном в Qt интерпретаторе. Она является обычным `QML` виджетом, принимающим команды от пользователя и передающим их в движок по обработке команд. Там происходит распознавание команды и передача ее параметров соответствующей функции-обработчику, в которой происходит непосредственное распознавание. Большинство функций-обработчиков написаны на `JavaScript`, что позволяет для добавления или изменения команды всего лишь пересобрать проект, а не перекомпилировать все файлы.

Сеть Skype и API `SkypeKit` предоставляют огромные возможности для реализации. Это, в первую очередь, аудио и видео звонки и конференции. Также в этот список входят звонки на мобильные и стационарные телефоны, передача файлов и контактов, предоставление общего доступа к экрану компьютера и многое другое. В рамках данной работы были воплощены лишь некоторые из этих функций, а именно: авторизация, загрузка контакт-листа и списка конференции, принятие и отправка мгновенных сообщений. Кроме того был реализован язык команд, управление с помощью клавиатурных комбинаций и переработан интерфейс.

ЛИТЕРАТУРА:

1. The Bell Tolls for Telcos? URL: <http://www.telegeography.com/press/press-releases/2013/02/13/the-bell-tolls-for-telcos/index.html> (дата обращения: 10.04.2013)
2. Open Source, RedHat and Security. URL: https://www.redhat.com/f/pdf/RH_OsRHandSecurity.pdf (дата обращения: 12.04.2013)
3. H. Plotkin. What (and Why) you should know about open-source software // Harvard Business Review – 1998 – 1 декабря